AD-A119 960 NAVAL RESEARCH LAB WASHINGTON DC F/G 17/2

MILITARY MESSAGE SYSTEMS: REQUIREMENTS AND SECURITY MODEL.(U)

SEP 82 C E LANDWEHR, C L HEITMEYER

UNCLASSIFIED NRL-MR-4925 NL

1 OF 1

AD A
119960

END
DATE
FILMED
11-82
DTIC

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Memorandum Report 4925 — 2. GOVT ACCESSION NO. AD A119668	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) MILITARY MESSAGE SYSTEMS: REQUIREMENTS AND SECURITY MODEL	5. TYPE OF REPORT & PERIOD COVERED Interim report on a continuing NRL problem. 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) C. E. Landwehr and C. L. Heitmeyer	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, DC 20375	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 63526N; X0911CC; 75-0105-A-2
11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE September 30, 1982 13. NUMBER OF PAGES 22
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Message systems	Multi-level security
Computer security	Security model requirements
Software requirements	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Military systems that process classified information must operate in a *secure* manner, i.e., they must adequately protect information against unauthorized disclosure, modification, and withholding. A goal of current research in computer security is to facilitate the construction of *multilevel secure systems*, i.e., systems that protect information of different classifications from users with different clearances. Security models are used to define the concept of security embodied by a computer system. A single model,

(Continues)

20. ABSTRACT (Continued)

called the Bell and LaPadula model, has dominated recent efforts to build secure systems but has some deficiencies. We are developing a new approach to defining security models based on the idea that a security model should be derived from a specific application. To evaluate our approach, we are formulating a security model for a family of military message systems. This paper introduces the message system application, summarizes our approach to developing a family of secure message systems, presents our security model, and summarizes our plans.

CONTENTS

A

MILITARY MESSAGE SYSTEMS: REQUIREMENTS AND SECURITY MODEL

I. INTRODUCTION

A system is secure if it adequately protects information that it processes against unauthorized disclosure, unauthorized modification, and unauthorized withholding (also called denial of service). We say "adequately" because no practical system can achieve these goals without qualification; security is inherently a relative concept. A secure system is multilevel secure if it protects information of different classifications from users with different clearances; thus some users are not cleared for all of the information that the system processes.

In recent years, several security models have been formulated to describe the security concepts embodied by a computer system [Land81]. Ideally, the security model associated with a given system will enable users to understand how to operate the system effectively and implementors to understand how to build the system correctly. Further, certifiers may use the model to determine whether the concepts of the system's security are consistent with the relevant policies and directives and whether the implementation matches the concepts [Land81].

To date, a single security model, called the Bell and LaPadula model [Bell75, Feie77], has dominated efforts to build secure systems. A system that enforces this model can protect against the unauthorized disclosure of information. Unfortunately, a system that strictly enforces the Bell-LaPadula model is impractical: in real systems, users frequently need to perform operations that, although they do not violate security, do violate the constraints of the model. For example, a user may need to extract an UNCLASSIFIED paragraph from a CONFIDENTIAL document and use it in an UNCLASSIFIED document. A system that strictly enforces the Bell-LaPadula model would prohibit this operation.

Consequently, systems based on this model usually contain mechanisms that permit some operations that the model prohibits (e.g., the mechanisms for privileges and "trusted processes" in KSOS [McCa79], SCOMP [Bonn81], and SIGMA [Ames79]). The presence of such mechanisms makes it more difficult to determine the actual security policy enforced by the system and complicates the user interface.

To avoid these problems, we are taking a different approach. We believe that a security model should be derived from a specific application. To evaluate our approach, we are formulating a security model for a family of military message systems.

Defining an application-based security model is part of a larger effort whose goals are (1) to develop a disciplined approach to the production of secure systems and (2) to produce fully worked-out examples of a requirements document and a security model. In this paper, we introduce the message system application, summarize our approach to developing a family of secure message systems, describe our security model, and summarize our plans.

Manuscript submitted August 10, 1982.

II. MILITARY MESSAGE SYSTEMS: REQUIREMENTS AND SECURITY-RELATED ISSUES

In recent years, message handling in the Department of Defense (DoD) has been increasingly automated [Heit80]. While the primary purpose of military message systems is to process formal messages, i.e., official messages exchanged by military organizations, such systems may also handle informal messages, i.e., unofficial messages exchanged by individuals. Formal messages are transmitted over military networks, such as AUTODIN; their format and use is governed by military standards (e.g., [Nava79]). Examples of informal messages are those currently supported by several message systems (e.g., HERMES [Myer77]) available on the ARPA network.

Functional Requirements

Message system operations fall into three categories: operations on incoming messages, operations on outgoing messages, and message storage and retrieval [Heit80]. Operations in the first category permit a user to display and print messages he has received. Second category operations support the creation, editing, and transmission of outgoing messages. Message storage and retrieval operations allow users to organize messages into message files and to retrieve messages via single keys (e.g., message id) or combinations of keys (e.g., subject and date of message transmission). Typically, military systems that process formal messages require the same operations as systems that process informal messages plus a number of additional operations. Examples of these additional operations are distribution determination, action and information assignment, coordination, and release [Heit80].

Security Requirements

Each formal military message is composed of several fields, including To, From, Info, Date-Time-Group, Subject, References, Text, Security, and Precedence. A classification, such as UNCLASSIFIED or SECRET, is assigned to each field and to some subfields, e.g., the paragraphs of the Text field; further, the overall message has a classification that is at least as high as that of any field or subfield. For example, the Subject field of a message may be classified at a lower level than the message as a whole, or two paragraphs of the Text field may have different classifications.

In some data processing applications, users process information at a single security level for long periods of time. In contrast, message system users often need to handle data of several classifications during a single computer session. For example, a user who is displaying or creating a message may need to handle data at several security levels simultaneously; so does a user working with a message file whose entries are classified at more than one level. Also, a user may wish to display a SECRET message at the same time that he is composing an UNCLASSIFIED reply.

Military message systems are required to enforce certain security rules. For example, they must insure 1) that users cannot gain access to data for which they are not cleared and 2) that the classifications of data cannot be modified improperly. Unfortunately, most automated systems cannot be trusted to enforce such rules. The result is that many computer-based systems in DoD operate in "system-high" mode: each user is cleared to the level of the most highly classified information on the system. A consequence of system-high operation is that all data leaving the computer system must be classified at the system-high level until a reviewer assigns the proper classification.

A goal of current computer security research is to permit the construction of multilevel secure systems. Unlike systems that operate at system-high, multilevel secure systems do not require all users to be cleared to the level of the highest information processed. Moreover, information leaving such a system can be assigned its actual security level rather than the level of the most highly classified information in the system. Unlike a system that operates at system-high, a multilevel system can preserve the different classifications of information that it processes.

Status of Secure Message Handling in the DoD

Many message systems in DoD operate in system-high mode. Examples include the National Military Intelligence Center Support System (NMIC-SS) [NMIC80], a message system for intelligence analysts, and the NMIC Information Display System (NIDS), a system for command centers. Because information leaving these systems is classified at the system-high level, no outgoing message can be transmitted until message handling personnel have reviewed a hard copy and approved the message classification. Upon completion of this manual procedure, the message is eligible for transmission.

The SIGMA message system [Roth79, Stot79], an experimental system developed for the Military Message Experiment [Wils82], is designed to operate in multilevel mode. Although SIGMA was built on the nonsecure TENEX operating system, its user interface is designed as though it were running on a security kernel, i.e , a software package that can be mathematically verified to conform to a formal model of the DoD security policy. It was assumed that SIGMA's user interface would remain unchanged if SIGMA were rebuilt to operate with a security kernel. The security kernel would enforce the notion of security embodied by the Bell-LaPadula model.

To evaluate the military utility of interactive message systems, more than 100 military officers and staff personnel used SIGMA operationally for more than a year. Because SIGMA is designed for multilevel operation, an advantage of such operation was realized: information leaving the system (e.g., an outgoing formal message) was assigned its actual classification rather than the system-high classification.

However, requiring SIGMA to enforce the Bell-LaPadula model caused several problems:

1. Multilevel objects, while required by military message systems, are not recognized by the Bell-LaPadula model. Message systems need to perform operations on multilevel objects, such as messages and message files; the Bell-LaPadula model only recognizes atomic or single-level objects.

2. Downgrading, an operation required by message system users, is prohibited by the Bell-LaPadula model. In some cases, users of message systems need to be able to reduce the classification of information. The Bell-LaPadula model disallows such an operation.

3. Application-specific security rules are beyond the scope of the Bell-LaPadula model. Only authorized users are allowed to invoke certain message system operations (e.g., message release). Bell-LaPadula has no mechanism for deciding whether a given user is authorized to invoke a particular operation.

The SIGMA design ignores the first problem. As a result, SIGMA requires users to perform downgrading in situations where a downgrade should not be required. For example, a SIGMA user who wants to extract a CONFIDENTIAL paragraph from a document with an overall classification of SECRET needs (i) to copy the paragraph to a new document with an overall classification of SECRET and (ii) to invoke the downgrade operation on the new document (consisting of the CONFIDENTIAL paragraph) to reduce its classification to CONFIDENTIAL. To deal with the second problem, SIGMA developers defined a "trusted process" that could, under certain circumstances, violate the Bell-LaPadula model. To handle the third problem, the developers produced software, external to both the security kernel and to the trusted process, that performs the required checks. Thus the security policy enforced by SIGMA was distributed among a security kernel, a trusted process, and software external to both.

III. DEVELOPING SECURE MILITARY MESSAGE SYSTEMS: A NEW APPROACH

At the Naval Research Laboratory, we are producing a requirements document, a design, and two prototypes of future military message systems (MMSs). One of the two prototypes will be designed for a military environment where little message handling is required (e.g., a submarine) and as a result will perform very few functions. The other prototype will perform a larger set of functions. The intent of the two prototypes is to demonstrate the viability of our approach to building secure systems.

Our approach calls for: (1) application of the family methodology [Parn76, Parn79], (2) formulation of a security model tailored to the message system application, (3) use of rapid prototyping, and (4) formal or semiformal specification of message system requirements. We describe our current security model in Section IV; (1), (3), and (4) are discussed below.

Family Methodology

Review of the requirements of future military message systems suggests that one system will not suffice for all environments. The DoD will need several similar systems that share certain functions and enforce military security. Because they will operate in different environments, these systems will have important differences, e.g., in their user command languages, in the functions that they perform, in the operating systems that they use, and in the hardware on which they are implemented.

The design problem is to exploit the similarities of these svstems without unduly constraining the individual variations that make each message system suitable for its particular environment. A message system for intelligence analysts, for example, shares certain features with a message system for command center personnel, but it may be unreasonable to insist that intelligence analysts and command center personnel use exactly the same system.

To deal with this problem, we have adopted the family methodology. This methodology requires developers to consider the entire family before building a single member. We are applying the methodology to two major products: (1) a requirements document that applies to all members and (2) a design document that defines a modular structure suitable for all members. In the modular structure, the different features of family members are assigned to separate modules, so that different family members can be produced by replacing modules without changing the overall program structure [Parn72]. For example, two family members that are identical except for their user command languages will differ only in their implementations of the user command language module.

Rapid Prototyping

To evaluate our requirements document, we are building rapid prototypes [Heit82]. These will allow us to test the document for consistency, completeness, and precision. More specifically, the rapid prototypes should help to answer the following questions:

- How can the functional specifications be improved?
- How should the security model be refined?
- How can we more closely integrate the security model and the functional specifications?

Requirements Document

We are producing a single requirements document for the MMS family. Because its goal is to describe the shared features of family members, the document excludes decisions about features that differentiate members, e.g., decisions about user command languages, computer hardware, and operating systems. Included in the document are (1) a set of functional capabilities (i.e., user services), where each family member is associated with some subset, and (2) a security model that applies to all members. Our requirements document adheres to the guidelines in [Heni80]: it describes what is required without making design decisions and attempts to provide a precise, consistent, and complete description of the requirements.

A central feature of the document is its use of an Intermediate Command Language (ICL) [Heit81], a superset of all functional capabilities required by family members. The ICL is partitioned into several groups so that all ICL statements within a single group are associated with the same data type. All objects of a given data type exhibit the same user-visible behavior. Sample data types are messages, message files, message file directories, and filters (these contain criteria for message retrieval). ICL statements grouped with the data type 'message' include: CREATE_MSG, DISPLAY MSG, REPLY_MSG, UPDATE_MSG, and RELEASE_MSG. While some ICL statements retrieve and display information (e.g., DISPLAY_MSG), others modify information (e.g., UPDATE_MSG).

Each member of the family is associated with a subset of the ICL. Because differences in user command languages do not affect the ICL, two members with the same functional capabilities but different user command languages will be associated with the same ICL subset. Given two systems where one system performs only a subset of the functions performed by the second, the ICL subset for the smaller system will be contained in the ICL subset for the larger system.

The mapping between a user command and an ICL statement will usually be one-to-one. However, in some cases, a single user command may lead to several ICL operations. To clarify our use of the ICL, we provide two appendices. Appendix A gives specifications for three ICL statements; Appendix B presents a sample user session with a message system belonging to the MMS family.

IV. SECURITY MODEL

The security model for the MMS family is intended to provide a framework for users to understand system security, to guide the design of each family member, and to provide a basis for certifiers to review the system. Although we intend to have a single security model for the entire MMS family, each member will require a separate security analysis. The model presented here is informal; we expect it to provide a basis for a more formal version that may be used as a basis for program verification efforts.

In this section we define some terms, use them to specify a model of how a user views the system's operation, and state assumptions and assertions, based on the terms and the model of operation, that are intended to be sufficient to assure the security of the system. The security model includes the definitions, user's view of operation, the assumptions, and the assertions. It is a revision of earlier work [Land80].

This model does not address auditing, although message systems clearly require auditing mechanisms. The existence of an audit trail may deter potential penetrators, but auditing is primarily a technique for detecting security violations after the fact. The security model focuses on assertions that, if correctly enforced, will prevent security violations. Consequently, assertions and assumptions about auditing do not appear; in a more detailed system specification, auditing requirements would be explicit.

Definitions

The definitions below correspond in most cases to those in general use and are given here simply to establish an explicit basis for the model. We distinguish between "objects", which are single-level, and "containers", which are multilevel. We also introduce the concept of "user roles", which correspond to particular job-related sets of privileges.

Classification: a designation attached to information that reflects the damag that could be caused by unauthorized disclosure of that information. A classification includes a sensitivity level (UNCLASSIFIED, CONFIDENTIAL, SECRET, or TOP SECRET) and a set of zero or more compartments (NATO, NUCLEAR, etc.). The set of classifications, together with the relation defining the allowed information flows between levels, form a lattice [Denn76]. Most dissemination control such as NATO only, NOFORN, and NOCONTRACTOR, can be handled as additional compartment names.

Clearance: the degree of trust associated with a person. This is established on the basis of background investigations and the functions required of the individual. It is expressed in the same way as classifications are, as a sensitivity level and a (possibly null) compartment set. In a secure MMS, each user will have a clearance, and functions performed by the MMS for that user may check the user's clearance and the classifications of objects to be operated on. Some other characteristics of a user, such as his nationality and employer, may also be treated as part of his clearance so that dissemination controls are handled properly within this framework.

UserID: a character string used to denote a user of the system. To use the MMS, a person must present a userID to the system, and the system must authenticate that the user is the person corresponding to that userID. This procedure is called logging in. Since clearances are recorded on the basis of one per userID, each user should have a unique userID.

User: A person who is authorized to use the MMS.

Role: The job the user is performing, such as downgrader, releaser, distributor, etc. A user is always associated with at least one role at any instant, and the user can change roles during a session. To act in a given role, the user must be authorized for it. Some roles may be assumed by only one user at a time (e.g., distributor). With each role comes the ability to perform certain functions.

Object: an abstraction implemented by an MMS. An object is the smallest unit of information in the system to which a classification is explicitly attached. An object thus contains no other objects -- it is not multilevel. There are many kinds of objects; an example is the date-time-group of a message.

Container: an abstraction implemented by an MMS. A container has a classification and may contain objects (each with its own classification) and/or other containers. In most MMS family members, message files and messages are containers. Some fields of a message (such as the Text field) may be containers as well. The distinction between an object and a container is based on type, not current contents: within a family member, if an entity of type message file is a container, then all message files in that family member are containers, even if some of them are empty or contain only objects and/or containers classified at the same level as the message file itself. Devices such as disks, printers, tape drives, and users' terminals will be containers, rather than objects, in most MMS family members.

Entity: either a container or an object.

Container Clearance Required (CCR): an attribute of some containers. For some containers, it is important to require a minimum clearance, so that if a user does not have at least this clearance, he cannot view any of the entities within the container. Such containers are marked with the attribute "Container Clearance Required" (CCR). For example, a user with only a CONFIDENTIAL clearance could be prohibited from viewing just the CONFIDENTIAL paragraphs of a message classified TOP SECRET. On the other hand, given a message file containing both TOP SECRET and CONFIDENTIAL messages, it may be acceptable to allow the user in question to view the CONFIDENTIAL ones, even though the container (message file) as a whole is classified TOP SECRET.

ID: identifier. An ID names an entity without referring to other entities. For example, the originator and date-time-group of a message constitute an ID for that message. Some, but not necessarily all, entities are named by identifiers. Entities may also be named in other ways, e.g., "the third paragraph in the text of the second message in the container INBOX."

Direct reference: a reference to an entity is direct if the entity's ID is used to name it.

Indirect reference: a reference to an entity is indirect if a sequence of two or more entity names (of which only the first may be an ID) is used to name it.

Operation: a function that can be applied to an entity. It may simply allow that entity to be viewed (e.g., display a message), or it may modify the entity (update a message), or both (create a message). Some functions may involve more than one entity (copy a message from one message file to another).

Access Set: a set of pairs (userID or role, operation) that is associated with an entity. The operations that may be specified for a particular entity depend on the type of that entity. For messages, operations include DISPLAY, UPDATE, DELETE, etc. The existence of a particular pair in the access set implies that the user corresponding to the specified userID or role is authorized to invoke the specified operation on the entity with which the set is associated.

Message: a particular type implemented by an MMS. In most MMS family members, a message will be a container, though messages may be objects in some receive-only systems. A message will include To, From, Date-Time-Group, Subject, and Text fields, and additional fields as well. A draft message also includes Drafter and Releaser fields.

User's View of MMS Operation

We present the following as a model of the use of a secure MMS. Terms defined above are printed in upper case.

People initiate use of the system by logging in. To log in, a person presents a USERID and the system performs authentication, using passwords, fingerprint recognition, or any appropriate technique. Following a successful authentication, the USER invokes OPERATIONS to perform the functions of the message system. The OPERATIONS a USER may invoke depend on his USERID and his current ROLE; by applying OPERATIONS, the USER may view or modify OBJECTS or CONTAINERS. The system enforces the security assertions listed below (that is, it prevents the user from performing OPERATIONS that would contradict these assertions).

Security Assumptions

It will always be possible for a valid user to compromise information to which he has legitimate access. To make the dependence of system security on user behavior explicit, we list the following assumptions. These assumptions are really security assertions that can only be enforced by the users of the system.

A1. The System Security Officer (SSO) is assumed to assign clearances, device classifications, and roles properly.

A2. The user is assumed to enter the correct classification when composing, editing, or reclassifying information.

A3. Within a classification, the user is assumed to address messages and to define access sets for entities he creates so that only users with a valid need-to-know can view the information.

A4. The user is assumed to control properly information extracted from containers marked CCR (i.e., to exercise discretion in moving that information to entities that may not be marked CCR).

The basis for these assumptions is that when there is no other source of information about the classification of an entity or the clearance of a person, the user is assumed to provide information that is correct.

Security Assertions

The following statements are to be demonstrated to hold for a multilevel secure MMS:

Authorization 1. A user can only invoke an operation on an entity if the user's userID or current role appears in the entity's access set along with that operation.

Classification hierarchy 2. The classification of any container is always at least as high as the maximum of the classifications of the entities it contains.

Changes to objects 3. Information removed from an object inherits the classification of that object. Information inserted into an object must not be classified at a level above the classification of that object.

Viewing 4. A user can only view (on some output medium) an entity with a classification less than or equal to the user's clearance and the classification of the output medium. (This assertion applies to entities referred to either directly or indirectly).

Viewing CCR entities 5. A user can view an indirectly referenced entity within a container marked "Container Clearance Required" only if the user's clearance is greater than or equal to the classification of that container.

Translating indirect references 6. A user can obtain the ID for an entity that he has referred to indirectly only if he is authorized to view that entity via that reference.

Labeling requirement 7. Any entity viewed by a user must be labelled with its classification.

Clearance-setting 8. Only a user with the role of System Security Officer can set the clearance recorded for a userID.

Downgrading 9. No classification marking can be downgraded except by a user with the role of downgrader who has invoked a downgrade operation.

Releasing 10. No draft message can be released except by a user with the role of releaser. The userID of the releaser must be recorded in the "releaser" field of the draft message.

Discussion

The purpose of this section is to make the effects of the model more apparent in particular cases. The paragraphs below are not part of the model; the previous section defines the model completely. Here we seek only to show how the model applies in specific circumstances.

1. What prevents a user from copying a classified entity to an unclassified entity?

   The classification of the entity being copied accompanies the data. Moving explicitly classified data to an unclassified entity is a violation of assertions 2 and 9 (unless the user requesting the operation is the downgrader and is performing a downgrade operation) since the classification of the data in question is effectively changed by the operation. Manipulations that affect only objects are covered by assertion 3.

2. What about copying a part of an object into another object?

   A part of an object inherits the classification of the whole object (assertion 3). Thus moving part of an object into another object is disallowed by assertions 2 and 3 unless classification of the former object is less than or equal to that of the latter. Note that this constraint does not affect the user's ability to remove an UNCLASSIFIED paragraph (an object) from a CONFIDENTIAL document (a container) and use it in an UNCLASSIFIED document (another container).

3. Does a user have a "login level" ?

   Login level is not explicitly part of the model, but the effect of a login level can be obtained through the classification of the user's terminal. The classification of the terminal is an initial upper bound on the classification of information that can be displayed on it (assertion 4). If the user wishes to restrict further the level of the information that appears on the terminal, he may invoke an operation to reduce the classification of the terminal. The right to determine the classification of shared devices (disks, printers, etc.) will generally belong to the SSO. Note that restricting the level of the information that can appear on the user`s terminal does not necessarily restrict the level of information that programs he invokes may have access to.

4. Processes do not appear in the model but surely will be present in the implementation. How will their activities be constrained?

   Operations, rather than processes or programs, are in the model because they correspond more closely to the user's view of the svstem. To the user, the system offers functions that may be invoked by tvping strings of characters, pushing function keys, etc. Each function can be understood by the user as an operation. In the implementation, processes are constrained to invoke only operations that preserve the truth of the assertions.

5. Which entities in a particular message system will be containers and which will be objects?

This decision is part of the next more detailed level of the stated model. Some likely choices are that messages and message files will be containers and that the date-time group will be an object. It is not necessary that all message systems in the family make the same choices. If two message systems that make different choices communicate, some method of mapping between those entities that are objects in one system and containers in the other must be defined.

6. How are entities created?

For each type of entity that users can create, there will be an operator that, when invoked, creates a new instance of that type. As with all other operations, only users who are authorized for it can invoke it. Thus, it is not necessarily the case that any particular user will be able to create any particular kind of object; he must be authorized to do so.

7. How does a user refer to an object or a container?

Some entities have identifiers (IDs) that allow them to be named directly. For example, the originator and date-time-group of a message constitute an ID for that message. A given entity may have zero, one, or more IDs. An entity may also be referred to indirectly by a qualified name (for example, "the third object in the container called 'MsgFile1'"). A user (or a program he invokes) can refer to an entity using any valid ID or qualified name (for example, "the third object in the container called "MsgFile1"). The former is called a direct reference and the latter an indirect reference.

8. What policy governs access to an entity in a container? (Is the classification of the container or of the contents tested, and with what is it compared)?

The answer to this question depends on the type of access (the operation invoked) and whether the reference is direct or indirect. If the entity is referred to directly for viewing, assertion 4 gives the appropriate restriction. If the reference is indirect, there are two cases depending on whether or not the object is within a container marked CCR. If it is, both assertions 4 and 5 have an effect, otherwise, only assertion 4 is relevant. Note that viewing an entity in a CCR container may be permitted if the reference is direct, while viewing the same entity via an indirect reference is disallowed. This provides a means for dealing with the aggregation problem without requiring duplicate copies of protected information.
If the operation has effects other than the viewing of objects, the other assertions may impose constraints. Assertion 1 always requires that the user (or his role) be in the access set for the entity-operation pair specified.

9. Is there anything in the system that is not (or is not part of) an entity?

From the user's point of view, no. There may be structures in the implementation that the user is unaware of and that would be difficult to assign a legitimate classification to (such as internal operating system queues, perhaps). Anything the user can create, display, or modify, however, must be (or be part of) an entity.

10. What are the relationships among a user, an operation he invokes, and programs that the operation may invoke on his behalf? (For example, what privileges do the programs inherit, how is it determined whether a given invocation is allowed under the security policy?)
A user has a clearance recorded in the system. When a user invokes an operation on an entity, his clearance and role, the appropriate device classifications, and the access set for that entity, determine whether he can invoke the operation. The user's ID or current role must be paired with the specified operation in the access set of the entity in question (assertion 1). If the operation allows information to be viewed via a given device, then the user's clearance and the classification of the output device must equal or exceed the classification of the information (assertion 4). Similarly, other security assertions must not be violated by the programs invoked as part of the requested operation.

11. There are no integrity levels or controls defined in the model. What prevents accidental/malicious modification of sensitive data?

The reasons for omitting integrity levels have been discussed separately [Land82]. The alteration of clearance or classification data is covered in the given set of assertions. Any alteration of data must presumably be accomplished by a user's invoking an operation; his authorization to invoke that operation is required by assertion 1. In the future, specific cases may be treated in additional assertions similar to assertion 10.

V. PLANS

Our plans include (1) refinement of the security model and the ICL and (2) construction of two prototype family members.

Refinements to the ICL and Security Model

The difference between the security model described in Section V and earlier versions is the result of applying the model to a particular ICL subset and observing the effect of each assertion on each ICL statement. In addition to changes in the model, this process has also produced changes (mostly additions) to the ICL. For example, we have added ICL statements that support the activities of the System Security Officer.

We expect this interaction between the security model and functional requirements to continue and to lead to further refinements. In particular, we expect to add assertions to the model concerning which users can invoke specific operations and thereby modify various data structures. We also plan to create a more formal and abstract version of the model to determine (1) whether it is *complete*, i.e., whether the concept of security that it embodies satisfies the legal requirements and includes all relevant message system functions; (2) whether it is *consistent*, i.e., whether it includes any contradictions; and (3) whether it is redundant (the informal model almost certainly will be).

Construction of Two Prototypes

We will use ICL subsets to specify the two MMS prototypes. We will build rapid prototypes to evaluate the reasonableness of both the ICL specifications and of the security model. Then, we plan to develop software designs for full-scale prototypes that implement the required functions and the security model. Finally, we intend to produce the full-scale prototypes according to the documented design.

VI. ACKNOWLEDGMENTS

Many individuals contributed to the work reported here. Discussions with D. Parnas led to an initial version of the security model. Later revisions of the model were based on reviews by S. Wilson, M. Cornwell, J. Miller, M. Schaefer, J. Millen, and others too numerous to mention. Participants in the 1982 Air Force Summer Study on Multi-Level Data Management Security also provided many helpful comments. For providing the funding that allows us to continue our work, we are grateful to H. O. Lubbes of the Naval Electronics Systems Command and to the Office of Naval Research.

REFERENCES

Ames78 S. R. Ames, Jr., and D. R. Oestreicher, "Design of a message processing system for a multilevel secure environment," in Proc. 1978 Nat. Comput. Conf., June 5-8, 1978, pp. 765-771.

Bell75 D. E. Bell and L. J. LaPadula, "Secure computer system: Unified exposition and multics interpretation," M74-244, MITRE Corp., Bedford, MA, July, 1975.

Bonn81 C. H. Bonneau, "Secure communications process kernel software, detailed specification, Part I, Rev. G," Honeywell, Inc., Avionics Division, St. Petersburg, FL, 1981.

Denn76 D. E. Denning, "A lattice model of secure information flow," Comm. Ass. Comput. Mach., vol. 19, no. 5, pp. 236-243, May 1976.

Feie77 R. J. Feiertag, K. N. Levitt, and L. Robinson, "Proving multilevel security of a system design," in Proc. 6th ACM Symp. Operating Systems Principles, ACM SIGOPS Operating System Rev., pp. 57-65.

Heit80 C. L. Heitmeyer and S. H. Wilson, "Military message systems: Current status and future directions," IEEE Trans. Commun., Vol. COM-28, No. 9, pp. 1645-1654.

Heit81 C. L. Heitmeyer, "An intermediate command language for military message systems," NRL Tech. Memo, Nov. 1981.

Heit82 C. L. Heitmeyer, C. E. Landwehr, and M. R. Cornwell, "The use of quick prototypes in the secure military message systems project," in Proc. ACM SIGSOFT Second Software Eng. Symp.: Workshop on Rapid Prototyping, Apr. 19-21, 1982.

Heni80 K. L. Heninger, "Specifying software requirements for complex systems: New techniques and their application," IEEE Trans. Software Eng., Vol. SE-6, pp. 2-13, Jan. 1980.

Land80 C. E. Landwehr, "Assertions for verification of multilevel secure military message systems," ACM Software Eng. Notes, Vol. 5, No. 3, pp. 46-47, July 1980.

Land81 C. E. Landwehr, "Formal models for computer security," ACM Computing Surveys, Vol. 13, No. 3, pp. 247-278, Sep. 1981.

Land82 C. E. Landwehr, "What security levels are for and why integrity levels are unnecessary," NRL Tech. Memo, Feb. 1982.

McCa79 E. J. McCauley and P. J. Drongowski, "KSOS - The design of a secure operating system," in *Proc. AFIPS Nat. Comput. Conf.*, June 4-7, 1977, pp. 345-353.

Mili81 J. S. Miller and R. G. Resnick, "Military message systems: Applying a security model," in *Proc. IEEE 1981 Symp. on Security and Privacy*, IEEE Cat. No. 81CH1629-5, pp. 101-112, Apr. 1981.

Mill82 J. K. Millen, personal communication, May 1982.

Myer77 T. H. Myer and C. D. Mooers, "HERMES user's guide," Bolt, Beranek and Newman, Inc., Cambridge, MA, Feb. 1977.

Nava79 "Naval telecommunications procedures telecommunications users manual," NTP 3, Naval Telecommun. Command, Washington, DC, July 1979.

NMIC80 "NMIC Support System User Manual," Planning Research Corp./Inform. Sci. Co., McLean, VA, Rep. PRC-R-2028, Apr. 1980.

Parn72 D. L. Parnas, "On the criteria to be used in decomposing systems into modules," *Commun. Ass. Comput. Mach.*, vol. 15, Dec. 1972.

Parn76 D. L. Parnas, "On the design and development of program families," *IEEE Trans. Software Eng.*, Vol. SE-2, pp. 1-9, Mar. 1976.

Parn79 D. L. Parnas, "Designing software for ease of extension and contraction," *IEEE Trans. Software Eng.*, Vol. SE-5, pp. 128-138, Mar. 1979.

Roth79 J. Rothenberg, "SIGMA message service: Reference manual," version 2.3, USC/Inform. Sci. Inst., Marina del Rey, CA, Rep. ISI/TM-78-11.2, June 1979.

Stot79 R. Stotz, R. Tugender, and D. Wilczynski, "SIGMA - An interactive message service for the military message experiment," in *Proc. 1979 Nat. Comput. Conf.*, June 4-7, 1979, pp. 855-861.

Wils82 S. H. Wilson, N. C. Goodwin, E. H. Bersoff and N. M. Thomas III, "Military message experiment - vol. I executive summary," Naval Res. Lab, Washington, DC, Rep. 4454, Mar. 1982.

APPENDIX A

Given below are specifications for three ICL commands associated with the data type 'message'. Each command has a name, input parameters, and a return value. The name is of the form XXX_YYY, where XXX is a mnemonic for the operation and YYY is an abbreviation for the associated data type; except for the underscore, each name is alphabetic and consists of all capitals. The names of input parameters and return values are lower-case and have been chosen to suggest the data type of the parameter or value. The Description provides the command semantics. Eventually, we intend to formalize the ICL semantics.

DATA TYPE: message (MSG)

Name:	CREATE_MSG	Input Pars:	msg_type sec_level	Returns:	msg_id

Description: Creates a draft message with a classification of sec_level and of type msg_type. Returns the id of the new message. The new message is assigned both a user-visible id (e.g., if msg_type is "formal," the message will be assigned a Date-Time-Group) which is recorded in the ID field of the message. The Security field of the message will be assigned the value sec_level. The drafter's name will be recorded in the Drafter field. Other fields of the message, such as Subject and Text, will be empty.

Name:	UPDATE_MSG	Input Pars:	msg_id msg	Returns:	-

Description: Sets the value of the message with id msg_id to msg. This command permits the user to make changes to a draft message.

Name:	GET_MSG	Input Pars:	msg_id	Returns:	msg

Description: Returns a copy of the message named msg_id. The message is not open for editing. The purpose of this command is to allow a user to display a message at his terminal.

APPENDIX B

Presented below is a sample user session with an MMS. Its purpose is to illustrate the kinds of operations an MMS will perform and the relationship between user commands and ICL statements. The user command language provided in this example is only one of many possible user command languages. The first column shows each user command, while the second column indicates the ICL statement associated with the user command; for simplicity, no ICL parameters are given. The third column shows the system response. Responses that are enclosed in percent signs are comments, while responses that are not bracketed are literal responses. Note that message files contain message citations, which in turn contain only part of the information in a message, e.g., the classification, the date received, the originator, and the subject. The citation also contains a pointer to the message.

```
User Command    ICL          System Response

displ mf new    GET_MF       message file:  inbox    classif:  TOP SECRET

                             5 U   6 Mar   denicoff at USC-ISI  Re: no cost ext
                             6 S   6 Mar   J.MILLER at BBNG     What's up?

displ cur msg   GET_MSG      %msg attached to citation 5 is displayed in full%

delete mc 5     DELETE_MC    done   %msg citation 5 is now marked deleted%

displ nxt msg   GET_MSG      %msg attached to citation 6 is displayed in full%

create secret
   mf millers   CREATE_MF    SECRET file named "millers" has been created

move mc 6
   millers      COPY_MC      done
                DELETE_MC

displ mf        GET_MF       message file:  inbox    classif:  TOP SECRET

                             1 S   2 Feb   landwehr            security options
                             2 U   3 Mar   denicoff at USC-ISI research option
                             3 T   4 Mar   reagan at USC-ISI   supply-side
                             4 U   5 Mar   bush                voodoo economics
                            *5 U   6 Mar   denicoff at USC-ISI Re: no cost ext
                            *6 S   6 Mar   J.MILLER at BBNG    What's up?

delete mc 2     DELETE_MC    done

expunge mf      EXPUNGE_MF   done
```

```
User Command      ICL            System Response

displ mf          GET_MF         message file:  inbox   classif:  TOP SECRET

                                 1 S   2 Feb   landwehr              security options
                                 2 T   4 Mar   reagan at USC-ISI     supply-side
                                 3 U   5 Mar   bush                  voodoo economics

displ dir         GET_MFDIR      message file directory:  heitmeyer

                                 inbox     TOP SECRET
                                 millers   SECRET
                                 voodoo    CONFIDENTIAL

get mf millers    none           done  %millers is now current file%

displ mf          GET_MF         message file:  millers   classif:  SECRET

                                 1 S   6 Mar   J.MILLER at BBNG     What's up?
```

END

DATE
FILMED

11-82

DTIC